Amendments to the Claims

This listing of claims will replace all prior versions, and listings of claims in the application.

Claims 1-22. (Currently cancelled).

23. (New) A superscalar microprocessor system capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor system comprising:

(a) an instruction fetch unit configured to fetch instructions from an instruction store and configured to provide a plurality of the instructions to an instruction buffer;

(b) an execution unit, coupled to the instruction fetch unit, configured to execute the plurality of the instructions from the instruction buffer in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of program order so that a load request can be made before a memory request corresponding to an instruction that precedes a load instruction corresponding to the load request in the program order and store requests in program order, the load store unit including:

(i) an address path adapted to manage load and store addresses and to provide the load and store addresses to the memory system;

(ii) load dependency check circuitry, wherein the load store unit does not make a particular load request when the load dependency check circuitry detects an address collision or write pending for that particular load request; and

(iii) a data path adapted to transfer data from the memory system to the execution unit in response to load requests, the data path configured to align data returned from the memory system to thereby permit data falling on a word boundary to be returned from the memory system to the execution unit in correct alignment,

wherein the superscalar microprocessor initiates more than one instruction in the one or more instructions in a clock cycle.

24. (New) The system according to claim 23, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests, wherein an address for a load request may be generated out-of-order.

25. (New) The system according to claim 23, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests, wherein an address for a store request may be generated out-of-order.

26. (New) A superscalar microprocessor system capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor system comprising:

(a) an instruction fetch unit for fetching instructions from an instruction store and for providing a plurality of the instructions to an instruction buffer;

(b) an execution unit, coupled to the instruction fetch unit, for executing the plurality of the instructions from the instruction buffer in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of program order so that a load request can be made before a memory request corresponding to an instruction that precedes a load instruction corresponding to the load request in the program order, the load store unit having,

(i) an address generation unit configured to generate load and store addresses for instructions in the instruction buffer, wherein a load address may be generated out of program order;

(ii) an address path adapted to manage the generated load and store addresses and to provide the generated load and store addresses to the memory system; and

(iii) a data path for transferring load data from the memory system to the execution unit,

wherein the superscalar microprocessor initiates more than one instruction in the one or more instructions in a clock cycle.

27. (New) The system according to claim 26, wherein the data path includes alignment control circuitry for generating a plurality of memory requests in response to a single

instruction in the plurality of the instructions when an operand of the single instruction falls on a word boundary.

28. (New) The system according to claim 27, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

29. (New) The system according to claim 27, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

30. (New) The system according to claim 26, wherein the load store unit comprises dependency check circuitry adapted to detect store-to-load dependencies, wherein the store-to-load dependency detection determines when data for a load request depends on a store request.

31. (New) The system according to claim 30, wherein the dependency check circuitry includes address comparison logic for comparing an address of a load request and an address of a store request.

32. (New) The system according to claim 30, wherein the dependency check circuitry includes relative age determining logic for determining the relative age of a load request and a store request.

33. (New) A computer system, comprising:

(a) a memory system that is configured to store instructions and data;

(b) a superscalar processor, connected to the memory system, for executing the instructions, wherein the superscalar microprocessor initiates more than one instruction in a clock cycle, the processor having,

(1) an instruction fetch unit for fetching instructions from the memory system and for providing a plurality of the instructions to an instruction buffer;

(2) an execution unit, coupled to the instruction fetch unit, for executing the plurality of the instructions from the instruction buffer in an out-of-order fashion, the execution unit including,

(i) a register file; and

(ii) a load store unit adapted to make load requests and store requests to the memory system, the load store unit adapted to make at least one load request out of the program order so that a load request can be made before a memory request corresponding to an instruction that precedes a load instruction corresponding to the load request in the program order, the load store unit further adapted to return data falling on a word boundary in correct alignment to the register file.

34. (New) The system according to claim 33, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load requests and store requests, wherein an address for a load request is generated out-of-order.

35. (New) The system according to claim 33, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load requests and store requests, wherein an address for a store request is generated out-of-order.

36. (New) The system according to claim 33, wherein the load/store unit includes alignment control circuitry for generating a plurality of memory requests in response to a single instruction in the plurality of the instructions when an operand of the single instruction falls on a word boundary.

37. (New) The system according to claim 36, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

38. (New) The system according to claim 36, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

39. (New) The system according to claim 33, wherein the load store unit comprises dependency check circuitry adapted to detect store-to-load dependencies, wherein the store-to-load dependency detection determines when data for a load request depends on a store request.

40. (New) The system according to claim 39, wherein the dependency check circuitry includes address comparison logic for comparing an address of a load request and an address of a store request.

41. (New) The system according to claim 39, wherein the dependency check circuitry includes relative age determining logic for determining the relative age of a load request and a store request.

42. (New) A superscalar microprocessor system capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor system comprising:

(a) an instruction fetch unit for fetching instructions from an instruction store and for providing a plurality of the instructions to an instruction buffer;

(b) an execution unit, coupled to the instruction fetch unit, for executing the plurality of the instructions from the instruction buffer in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of program order so that a load request can be made before a memory request corresponding to an instruction that precedes a load instruction corresponding to the load request in the program order, the load store unit having,

(i) an address generation unit configured to generate load and store addresses for instructions in the instruction buffer, wherein a load address is generated out of program order;

(ii) an address path adapted to manage the generated load and store addresses and to provide the generated load and store addresses to the memory system; and

(iii) dependency check circuitry adapted to detect store-to-load dependencies, wherein the store-to-load dependency detection determines when data for a load request depends on a store request; and

(iv) a data path for transferring load data from the memory system to the execution unit, the data path configured to align data returned from the memory system to thereby permit data falling on a word boundary to be returned from the memory system to the execution unit in correct alignment,

wherein the superscalar microprocessor initiates more than one instruction in the one or more instructions in a clock cycle.

43. (New) The system according to claim 42, wherein the data path includes alignment control circuitry for generating a plurality of memory requests in response to a single instruction in the plurality of the instructions when an operand of the single instruction falls on a word boundary.

44. (New) The system according to claim 43, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

45. (New) The system according to claim 43, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

46. (New) The system according to claim 42, wherein the dependency check circuitry includes relative age determining logic for determining the relative age of a load instruction in the plurality of the instructions and a store instruction in the plurality of the instructions.